

Sistema de Controle de Acesso Baseado em IoT com Integração a Aplicativo Móvel Utilizando RFID, ESP32 e Firebase

Hugo Felipe Rafael Muquiuti¹, Alessandro Paulo de Oliveira², Felipe Walter Dafico Pfrimer²

¹Universidade Tecnológica Federal do Paraná, Toledo-PR, Brasil
(hugomuquiuti@alunos.utfpr.edu.br)

²Universidade Tecnológica Federal do Paraná, Toledo-PR, Brasil

Resumo: Este trabalho apresenta o desenvolvimento de um sistema de controle de acesso IoT com ESP32, RFID e Firebase. Permite autenticação por tags e aplicativo, com gerenciamento remoto de usuários. Testes indicaram leitura confiável até 2 cm, tempo de resposta entre 250 e 600 ms e acionamento em cerca de 1 s. O sistema operou de forma estável, com comunicação eficiente mesmo sob variações no Wi-Fi, demonstrando viabilidade para aplicações de controle de acesso de baixo custo.

Palavras-chave: Internet das Coisas; RFID; Firebase; Controle de acesso.

INTRODUÇÃO

Nas últimas décadas, o avanço das tecnologias digitais e da Internet das Coisas (IoT) tem transformado a forma como dispositivos eletrônicos interagem entre si e com os usuários. Nesse paradigma, objetos físicos equipados com sensores, atuadores e capacidades de comunicação podem ser conectados à internet, permitindo a coleta, troca e processamento de informações em diferentes aplicações (Atzori et al., 2010). Esse conceito tem sido aplicado em diversas áreas, incluindo automação residencial, cidades inteligentes e monitoramento ambiental (Gubbi et al., 2013).

Entre as aplicações relacionadas à segurança, os sistemas de controle de acesso têm se destacado como uma importante área de desenvolvimento. Tradicionalmente, esses sistemas utilizam chaves mecânicas ou dispositivos eletrônicos simples, que apresentam limitações relacionadas à segurança, gerenciamento de usuários e registro de eventos. Com o avanço das tecnologias embarcadas, da conectividade sem fio e da Internet das Coisas (IoT), tornou-se possível desenvolver soluções mais avançadas que permitem o controle e o monitoramento de acessos por meio de dispositivos conectados à internet (Alaba et al., 2017).

Nesse contexto, a tecnologia RFID (*Radio Frequency Identification*) tem sido amplamente utilizada para identificação automática de pessoas e objetos. Sistemas baseados em RFID permitem que usuários sejam autenticados por meio de cartões ou etiquetas eletrônicas, tornando o processo de controle de acesso mais prático e seguro (Want, 2006). Quando combinada com microcontroladores com

conectividade Wi-Fi, como o ESP32, essa tecnologia possibilita a criação de dispositivos conectados capazes de se comunicar diretamente com serviços em nuvem e aplicações móveis.

Além disso, o uso de plataformas de computação em nuvem tem ampliado as possibilidades de desenvolvimento de sistemas IoT. Serviços em nuvem permitem o armazenamento de dados, sincronização em tempo real e gerenciamento remoto de dispositivos conectados, possibilitando a integração entre sistemas embarcados e aplicações móveis (Botta et al., 2016; Borgia, 2014). Em conjunto com ambientes de desenvolvimento multiplataforma, como o Flutter, essas tecnologias permitem a criação de aplicativos móveis capazes de interagir diretamente com dispositivos IoT de forma simples e eficiente.

Alguns trabalhos recentes têm investigado o uso de tecnologias de Internet das Coisas na integração entre dispositivos embarcados, redes de comunicação e aplicações conectadas, bem como aspectos relacionados à segurança, confiabilidade e operação em tempo real desses sistemas (Bertoli et al., 2023; Barral Vales et al., 2024). Nesse contexto, aplicações de controle de acesso baseadas em dispositivos embarcados e tecnologias de identificação automática, como RFID, têm sido exploradas como alternativas para sistemas de segurança inteligentes. Essas soluções combinam microcontroladores com conectividade à internet, tecnologias RFID, possibilitando o monitoramento e controle de acessos em tempo real (Surapaneni et al., 2024). Além disso, abordagens recentes têm investigado arquiteturas IoT integradas com múltiplos serviços e funcionalidades,

incluindo autenticação, monitoramento e controle remoto em tempo real, com foco em eficiência, escalabilidade e baixo custo (Hasib et al., 2026).

Nesse contexto, o desenvolvimento de sistemas de controle de acesso baseados em IoT representa uma alternativa interessante para a implementação de soluções de segurança modernas, especialmente quando associado ao uso de hardware acessível e tecnologias abertas. A integração entre dispositivos embarcados, serviços em nuvem e aplicações móveis possibilita a criação de sistemas capazes de oferecer funcionalidades como autenticação eletrônica, gerenciamento remoto de usuários e registro de eventos de acesso.

Assim, este trabalho tem como objetivo desenvolver e implementar um sistema de controle de acesso baseado em Internet das Coisas, por meio de uma fechadura eletrônica acionada por tag RFID ou por aplicativo móvel. Nesse contexto, a principal contribuição da proposta está na implementação e validação experimental de uma arquitetura de baixo custo que integra autenticação local e gerenciamento remoto em uma solução unificada. O sistema utiliza o microcontrolador ESP32 como unidade embarcada, responsável pela leitura das tags, comunicação com o Firebase Realtime Database e acionamento da fechadura, enquanto o aplicativo desenvolvido em Flutter permite a autenticação e o gerenciamento remoto de usuários autorizados em tempo real.

MATERIAL E MÉTODOS

O sistema de controle de acesso desenvolvido, apresentado na Figura 1, foi concebido com a integração de diferentes tecnologias, buscando oferecer uma solução prática, segura e de fácil utilização para o gerenciamento de entrada em ambientes. O fluxo de funcionamento do sistema, sob a perspectiva do usuário, é ilustrado na Figura 2. A proposta parte da ideia de centralizar o controle nas mãos do administrador, permitindo que todo o processo seja realizado de forma remota e intuitiva. A interação com o sistema acontece por meio de um aplicativo móvel desenvolvido em Flutter, que pode ser utilizado tanto em dispositivos Android quanto iOS. Por meio desse aplicativo, o administrador pode realizar o cadastro dos usuários, utilizando informações como e-mail e senha, além de definir quais deles possuem autorização para acessar o local monitorado. Os dados cadastrados são armazenados no Firebase, que atua como base central do sistema, garantindo que as informações estejam sempre disponíveis e atualizadas. Essa estrutura permite que as decisões de acesso sejam tomadas de forma rápida e consistente, independentemente da localização do usuário ou do administrador. Uma vez que o usuário recebe a autorização, o próprio aplicativo disponibiliza uma nova interface com a opção de

acesso. Ao selecionar essa opção, um comando é enviado ao sistema, acionando o mecanismo responsável pela abertura da fechadura. Dessa forma, o processo de entrada ocorre de maneira simples, sem a necessidade de chaves físicas, mantendo ao mesmo tempo um nível adequado de segurança e controle.

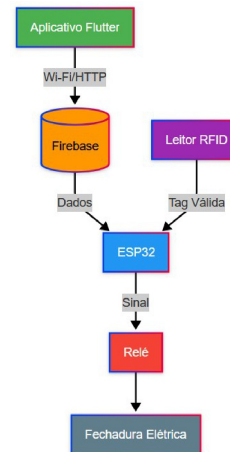


Figura 1. Diagrama de blocos do sistema.

No ambiente de acesso, o funcionamento do sistema ocorre de forma automática e integrada aos dados previamente cadastrados. Ao se aproximar do ponto de entrada, o usuário apresenta sua tag ao leitor RFID, que realiza a leitura do identificador único do dispositivo. Essa informação é então enviada ao microcontrolador ESP32, responsável pelo processamento das decisões do sistema. A partir desse momento, o ESP32 compara o identificador lido com os dados armazenados na base em nuvem, Firebase, verificando se a tag está associada a um usuário autorizado. Caso a identificação seja válida, o sistema envia um sinal para um módulo relé, que atua como um interruptor eletrônico. Esse acionamento permite energizar a fechadura elétrica, liberando o acesso do usuário de forma rápida e segura. Por outro lado, se a tag não estiver cadastrada ou não possuir autorização, o acesso é automaticamente negado.

O ESP32, responsável pela comunicação via Wi-Fi e integração com o Firebase, é conectado ao leitor RFID e ao módulo relé. O acionamento da trava solenoide é realizado indiretamente por meio do relé, enquanto a alimentação do sistema é provida por uma fonte externa, com adequação de tensão realizada por um conversor *step-down*, que fornece alimentação ao ESP32, o qual, por sua vez, disponibiliza tensão de operação para o módulo RFID, conforme apresentado na Figura 3. O programa do ESP32 foi desenvolvido na plataforma Arduino IDE, com o auxílio de bibliotecas específicas disponíveis em seu gerenciador de bibliotecas. Foram utilizadas bibliotecas voltadas à leitura de tags RFID, armazenamento de dados na memória, comunicação

via Wi-Fi e integração com o Firebase, permitindo a comunicação entre o microcontrolador, o aplicativo e a base de dados em nuvem.

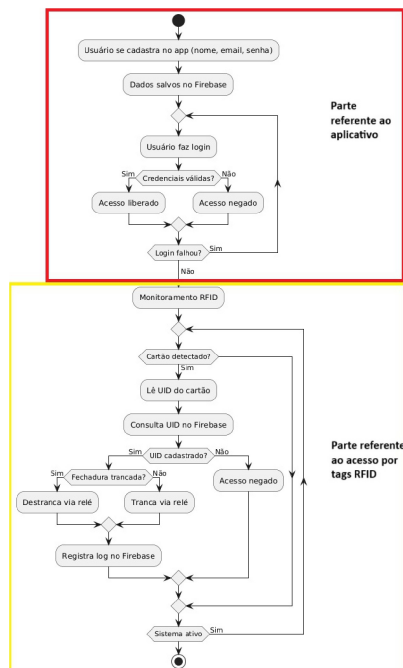


Figura 2. Fluxo de interação do usuário com o sistema.

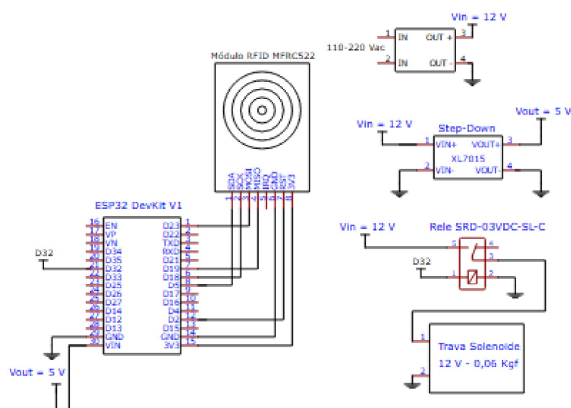


Figura 3. Arquitetura física do sistema de controle de acesso.

O acionamento do relé foi realizado diretamente por meio de um pino digital do ESP32 (D32). Durante o funcionamento do sistema, o programa desenvolvido para o ESP32 mantém o microcontrolador em estado de monitoramento contínuo, aguardando a aproximação de uma tag RFID. Quando uma tag é detectada, seu identificador único (UID) é lido e comparado com os dados armazenados no Firebase Realtime Database. Caso a tag esteja cadastrada e autorizada, o programa aciona o pino responsável pelo controle do relé, permitindo a liberação momentânea da fechadura. Em seguida, o evento de acesso é registrado no banco de dados, incluindo

informações como data, horário e identificação do usuário. Por outro lado, quando a tag não está cadastrada, o acesso é negado. Nessa situação, o UID da tag é exibido no monitor serial da Arduino IDE e também disponibilizado no Firebase Realtime Database. Isso permite que o administrador realize o cadastro da nova tag de duas formas distintas: diretamente no código do ESP32, conforme ilustrado na Figura 4, ou manualmente por meio da interface administrativa do Firebase.

```
// Cadastra tags
void cadastrarTagsTeste() {
  writeCardToEEPROM(0, "CA D7 F5 0F"); // Cadastra a primeira tag
  writeCardToEEPROM(1, "AA 43 F9 0F"); // Cadastra a segunda tag
  Serial.println("Tags de teste cadastradas na EEPROM.");
}
```

Figura 4. Cadastro de tags direto no código.

No modo de cadastro via Firebase Realtime Database, a inclusão de uma nova tag é realizada inserindo seu identificador no caminho correspondente às tags e atribuindo o valor "1", indicando autorização. Somente após essa configuração a tag passa a ser reconhecida como válida, permitindo a liberação do acesso por meio da abertura da fechadura (trava solenoide), conforme apresentado na Figura 5.

O programa desenvolvido para o ESP32 foi configurado para atuar como um servidor HTTP, disponibilizando um *endpoint* responsável por receber as requisições enviadas pelo aplicativo. Ao receber uma solicitação, o sistema realiza a verificação de sua validade e, caso esteja autorizada, executa o acionamento do relé, permitindo a liberação da fechadura. Essa abordagem possibilita o controle remoto do acesso de forma segura e eficiente, assegurando que apenas requisições autorizadas sejam processadas.

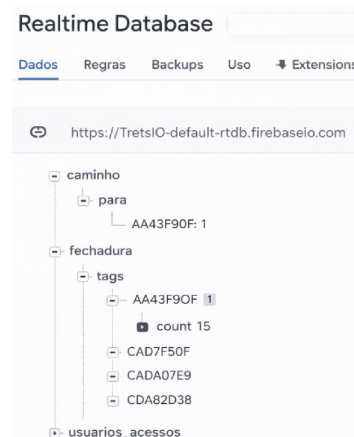


Figura 5 – Interface de cadastro de tags no Firebase Realtime Database.

O aplicativo foi desenvolvido utilizando o ambiente Flutter. Para sua implementação, foi utilizado o Visual Studio Code, em conjunto com o SDK do

Flutter e o gerenciador de pacotes Dart. Além disso, foi realizada a configuração do Firebase para autenticação e gerenciamento de usuários. As telas de login, registro e recuperação de senha foram desenvolvidas com integração ao Firebase Authentication, permitindo a autenticação por meio de e-mail e senha. Essas interfaces seguem um fluxo de navegação organizado, com validação dos dados inseridos e tratamento de possíveis erros, conforme ilustrado nas Figuras 6, 7 e 8. Na tela de login, Figura 6, o usuário pode acessar o sistema informando o e-mail e a senha previamente cadastrados. Após a autenticação, o acesso à funcionalidade de abertura da fechadura (trava solenóide) é disponibilizado. A interface também oferece as opções “Você ainda não possui uma conta?”, que direciona o usuário para a tela de registro, e “Esqueceu sua senha?”, que permite a redefinição da senha cadastrada. A tela de registro, Figura 7, permite o cadastro de novos usuários por meio da inserção de um e-mail válido e de uma senha com, no mínimo, seis caracteres. Após o envio das informações, o administrador do sistema pode, por meio do Firebase, aprovar ou recusar o cadastro. Nessa mesma interface, o usuário também pode retornar à tela de login ou acessar a opção de recuperação de senha. Na tela recuperação da senha, Figura 8, o usuário pode redefinir sua senha por meio do envio de um link de recuperação para o endereço de e-mail previamente cadastrado no sistema. Esse recurso pode ser utilizado tanto em situações em que a senha foi esquecida quanto quando o usuário deseja alterá-la por motivos de segurança. A partir do *link* enviado, é possível realizar a atualização da senha, restabelecendo o acesso ao aplicativo.

O aplicativo foi desenvolvido utilizando o pacote Provider para o gerenciamento de estado, permitindo a atualização eficiente das informações entre os diferentes componentes da aplicação. A autenticação dos usuários é realizada por meio do Firebase Authentication, possibilitando o login e o registro com e-mail e senha. Após a autenticação, os dados dos usuários são armazenados no Firebase Realtime Database, onde também é realizado o controle das permissões de acesso. O sistema conta com uma camada administrativa, na qual o administrador pode aprovar ou rejeitar manualmente novos cadastros diretamente no Firebase Realtime Database. Além disso, o sistema realiza o registro automático da data e do horário em que a fechadura é acionada por cada usuário autenticado, conforme ilustrado na Figura 9.

A Figura 10 apresenta a estrutura utilizada no Firebase, destacando o serviço Firebase Authentication, responsável pela autenticação de usuários por meio de e-mail e senha. Foram realizados testes de usabilidade e desempenho, com foco na responsividade da interface e na eficiência da comunicação com os serviços do Firebase. O sistema

de autenticação, baseado no Firebase Authentication, garante que apenas usuários autorizados tenham acesso às funcionalidades do aplicativo.

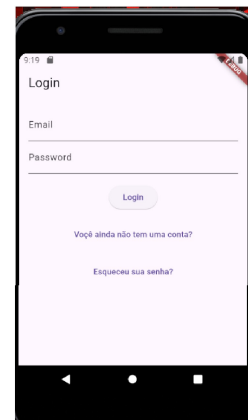


Figura 6. Tela de login.

Além disso, os testes avaliaram a integridade dos dados armazenados e a estabilidade da conexão com o Firebase Realtime Database em diferentes condições de uso. O Firebase Realtime Database desempenha um papel fundamental na estrutura do sistema, sendo utilizado para armazenar e gerenciar as informações relacionadas aos usuários e ao controle de acesso. Por meio dessa plataforma, são registrados os dados de cadastro, como o e-mail e o status de aprovação de cada usuário, definidos pelo administrador da aplicação, conforme ilustrado na Figura 11. Adicionalmente, o banco de dados mantém um histórico de eventos, registrando as datas e horários em que a fechadura foi acionada, o que permite um acompanhamento detalhado das atividades do sistema. Essa abordagem possibilita não apenas o gerenciamento em tempo real, mas também a escalabilidade e a sincronização eficiente entre o aplicativo e os dispositivos conectados, como o ESP32.

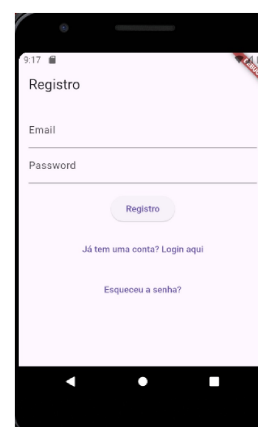


Figura 7. Tela de registro.

O programa desenvolvido para o ESP32 foi configurado para atuar como um servidor HTTP, disponibilizando um endpoint responsável por

receber os comandos enviados pelo aplicativo. Ao receber uma requisição, o sistema verifica sua validade e, caso esteja autorizada, aciona o relé responsável pela liberação da fechadura. Essa abordagem permite o controle remoto do sistema de forma prática e segura, garantindo que apenas solicitações autorizadas sejam processadas.

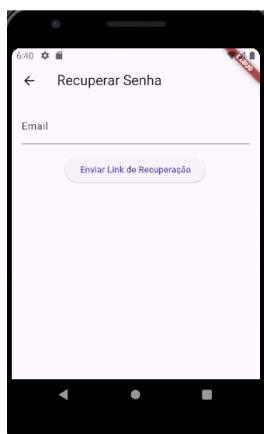


Figura 8. Tela recuperação de senha.

Após a finalização da montagem física do circuito, foram realizados testes com o objetivo de validar o funcionamento, a estabilidade e a confiabilidade do sistema como um todo. Para uma análise mais organizada, os testes foram conduzidos em etapas, contemplando a avaliação individual de cada subsistema.

Foram realizados testes com o módulo RFID MFRC522, variando a distância entre a tag e o leitor de 0 a 5 cm. Também foram realizados testes de repetição para avaliar a consistência das leituras e a comunicação SPI com o ESP32.

Foram realizados testes com o ESP32 em diferentes condições de rede Wi-Fi, incluindo ambientes domésticos, uso de roteador portátil e situações com sinal reduzido, com o objetivo de avaliar a robustez da comunicação com o Firebase Realtime Database. Durante os testes, foram simuladas interrupções na conexão, seguidas de processos de reconexão, permitindo verificar o tratamento de falhas e a retomada da comunicação. As operações de leitura e escrita no banco de dados foram acompanhadas por meio do console do Firebase, possibilitando a verificação do correto envio e recebimento das informações. Foram realizados testes com o relé tanto de forma isolada quanto integrado ao sistema, com o objetivo de avaliar seu desempenho no acionamento da fechadura. Além disso, foram conduzidos testes utilizando comandos remotos por meio do aplicativo desenvolvido em Flutter, simulando o acesso por diferentes usuários em horários distintos. Foram realizadas medições do intervalo de tempo entre a leitura de uma tag RFID

ou o envio de um comando pelo aplicativo até o acionamento físico do relé.



Figura 9. Aprovação do e-mail e sua data.

Authentication | with Identity Platform

Usuários | Método de login | Modelos | Uso | Configurações | Extensions

Identificador	Provedores	Data de criação	Último login	UID do usuário
email2r@gmail.com	✉	9 de abr. de 2025	9 de abr. de 2025	WRa0pbtloaZzepllyOGM4znWgcopimnfp...
email1@gmail.com	✉	9 de abr. de 2025	9 de abr. de 2025	gsizA4MDVQBpjuRdnn46bRUYc73

Figura 10. Autenticação no firebase.

Realtime Database

Dados | Regras | Backups | Uso | Extensions

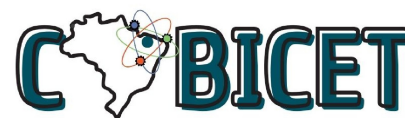
```

https://tcc-rfid-default-rtdb.firebaseio.com
{
  usuarios: {
    F0r9cpXdpNg9aA8tYPFypvTbXFU2: {
      email: "Email2@gmail.com"
      status: "pendente"
    },
    gsizA4MDVQBpjuRdnn46bRUYc73: {
      email: "Email1@gmail.com"
      status: "aprovado"
    }
  },
  usuarios_acessos: {
    email1@gmail.com: {
      "-ONQt2srVnay9tExHwJJ3": {
        horario: "2025-04-09 15:51:05"
        status: "aberta"
      }
    }
  }
}
  
```

Figura 11. Controle de acesso.

RESULTADOS E DISCUSSÃO

Leitura RFID com o módulo MFRC522: os testes realizados com diferentes cartões e tags RFID permitiram avaliar o desempenho do sistema de leitura em função da distância entre a tag e o leitor. Os resultados indicaram que a leitura ocorre de forma confiável em distâncias de até 2 cm. Para distâncias superiores, observou-se degradação significativa no desempenho, sendo que, a partir de aproximadamente 3 cm, a taxa de leitura reduz-se para cerca de 20%. Além disso, os testes de repetição evidenciaram alta confiabilidade do sistema dentro da faixa de operação, com taxa de acerto de 100% nas leituras realizadas a uma distância de até 2 cm. Não foram observadas falhas relevantes na comunicação SPI entre o módulo MFRC522 e o ESP32 durante os



ensaios. Esse comportamento está de acordo com as características do módulo MFRC522, descritas na literatura, que indicam curto alcance de operação, tipicamente limitado a poucos centímetros, devido às características da comunicação por acoplamento indutivo na frequência de 13,56 MHz. Dessa forma, os resultados obtidos confirmam o comportamento esperado do dispositivo, ao mesmo tempo em que demonstram sua adequação para aplicações de controle de acesso por proximidade, nas quais a leitura em curtas distâncias exige a aproximação intencional da tag, reduzindo leituras indevidas e contribuindo para maior segurança do sistema.

Conectividade Wi-Fi e integração com o Firebase: os resultados indicaram que a variação na qualidade do sinal Wi-Fi não compromete o funcionamento do sistema. Entretanto, em condições de sinal reduzido, observou-se um aumento no tempo de atualização do Firebase Realtime Database, com atrasos de alguns segundos. Por outro lado, na ausência total de conexão com a internet, as leituras realizadas não são registradas no banco de dados, não havendo alteração no estado da fechadura. Esse comportamento impede a liberação indevida do acesso, contribuindo para a segurança do sistema nessas condições. Esse resultado está de acordo com o modelo de funcionamento de sistemas baseados em serviços em nuvem, nos quais a disponibilidade da conexão é essencial para a sincronização de dados em tempo real. Dessa forma, embora a latência aumente em condições de sinal degradado, o sistema mantém seu funcionamento quando há conectividade disponível. Na ausência de conexão com a internet, observa-se um comportamento seguro, no qual a fechadura não é acionada, impedindo acessos não autorizados.

Acionamento do relé e resposta da fechadura: observou-se que, após a validação de uma tag autorizada, o relé é acionado e desativado em um intervalo de aproximadamente 1 segundo. Os resultados demonstraram que o relé respondeu corretamente em todas as situações avaliadas, garantindo o funcionamento adequado da abertura da fechadura. Esse comportamento indica que o sistema apresenta tempo de resposta compatível com aplicações de controle de acesso, nas quais pequenos atrasos não comprometem a interação do usuário. Além disso, a resposta consistente do relé evidencia a confiabilidade do acionamento tanto em operações locais quanto remotas.

Tempo de resposta e latência: os resultados indicaram que o tempo de resposta variou entre 250 e 600 milissegundos, apresentando variação conforme a qualidade da conexão Wi-Fi. Observou-se que esses valores não comprometeram o funcionamento do sistema durante os testes realizados. Esse desempenho é compatível com aplicações de controle de acesso, nas quais atrasos dessa ordem não

prejudicam a interação do usuário nem a eficiência do sistema. A influência da qualidade da conexão Wi-Fi no tempo de resposta evidencia a dependência da comunicação em nuvem, característica comum em sistemas baseados em IoT.

REFERÊNCIAS

- ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. *Computer Networks*, v. 54, n. 15, p. 2787–2805, 2010.
- GUBBI, J.; BUYYA, R.; MARUSIC, S.; PALANISWAMI, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, v. 29, n. 7, p. 1645–1660, 2013.
- ALABA, F. A.; OTHMAN, M.; HASHEM, I. A. T.; ALI, F. Internet of Things security: A survey. *Journal of Network and Computer Applications*, v. 88, p. 10–28, 2017.
- WANT, R. An introduction to RFID technology. *IEEE Pervasive Computing*, v. 5, n. 1, p. 25–33, 2006.
- BOTTA, A.; DE DONATO, W.; PERSICO, V.; PESCAPÉ, A. Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*, v. 56, p. 684–700, 2016.
- BORGIA, E. The Internet of Things vision: Key features, applications and open issues. *Computer Communications*, v. 54, p. 1–31, 2014.
- BERTOLI, G. C.; FERNANDES, G. V. C.; MONICI, P. H. B.; GUIBO, C. H. A.; PEREIRA JR., L. A.; SANTOS, A. Design and implementation of intelligent packet filtering in IoT microcontroller-based devices. *arXiv preprint, arXiv:2305.19214*, 2023.
- BARRAL VALES, V.; FERNÁNDEZ, O. C.; DOMÍNGUEZ-BOLAÑO, T.; ESCUDERO, C. J.; GARCÍA-NAYA, J. A. Fine time measurement for the Internet of Things: A practical approach using ESP32. *arXiv preprint, arXiv:2401.16517*, 2024.
- SURAPANENI, R. K.; HAMEED, S.; CHARAN, J. RFID-based elevator access control system. In: *Proceedings of the International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2024.
- HASIB, A.; AKIB, A. S. M. A.; ANKUR, N. D.; GIRI, A. Dual-modality IoT framework for integrated access control and environmental safety monitoring with real-time cloud analytics. *arXiv preprint, arXiv:2601.20366*, 2026.